WHAT IS CLAIMED IS:

1. A system for simulating machine instructions on a host machine comprising:

a monitor that translates the machine instructions into translated code and prevents the translated code from being modified;

a virtual machine that executes the translated code stored in memory; and

a kernel that detects exceptions occurring in the virtual machine and transfers control between the virtual machine and the monitor according to a type of the exceptions.

2. The system of claim 1 wherein the host operating system also supports a full platform simulator that includes device models.

3. The system of claim 1 wherein the translated code and the original machine instructions access the memory using the same addresses.

4. The system of claim 1 wherein the monitor further includes an auxiliary simulator that executes the machine instructions.

5. The system of claim 1 wherein the monitor replaces one of the machine instructions with a capsule if the machine instruction accesses a system state of a central processing unit of the host machine.

6. The system of claim 1 wherein the monitor modifies a descriptor table to prevent the translated code from being modified.

7. The system of claim 6 wherein the monitor modifies the descriptor table to remove a portion of the segment that overlaps with the memory storing the translated code.

8. The system of claim 6 wherein the monitor modifies the descriptor table to replace the segment with a substitute segment, which, when accessed, causes an exception to be generated.

9. A method of simulating machine instructions on a host machine comprising:

translating the machine instructions into translated code;

storing the translated code in memory;

executing the translated code;

preventing the translated code from being modified;

detecting exceptions in the execution of the translated code; and

transferring control to an appropriate simulation module on the host machine according to a type of the exceptions.

10. The method of claim 9 further comprising simulating a device.

11. The method of claim 9 further comprising accessing memory by the translated code using the same addresses as the addresses used by the original machine instructions.

12. The method of claim 9 further comprising replacing one of the machine instructions with a capsule if the machine instruction accesses a system state of a central processing unit of the host machine.

13. The method of claim 9 further comprising modifying a descriptor table to prevent the translated code from being modified, the descriptor table including attributes of a segment of the memory.

14. The method of claim 13 further comprising modifying the descriptor table to remove a portion of the segment that overlaps with the memory storing the translated code.

15. The method of claim 13 further comprising modifying the descriptor table to replace the segment with a substitute segment, which, when accessed, causes an exception to be generated.

16. A computer program product residing on a computer readable medium comprising instructions for causing the computer to:

translate the machine instructions into translated code;

store the translated code in memory;

execute the translated code;

prevent the translated code from being modified;

detect exceptions in the execution of the translated code; and

transfer control to an appropriate simulation module on the host machine according to a type of the exceptions.

17. The computer program product of claim 16 further comprising instructions for causing the computer to simulate a device.

18. The computer program product of claim 16 further comprising instructions for causing the computer to access

memory by the translated code using the same addresses as the addresses used by the original machine instructions.

19. The computer program product of claim 16 further comprising instructions for causing the computer to replace one of the machine instructions with a capsule if the machine instruction accesses a system state of a central processing unit of the host machine.

20. The computer program product of claim 16 further comprising instructions for causing the computer to modify a descriptor table to prevent the translated code from being modified, the descriptor table including attributes of a segment of the memory.

21. The computer program product of claim 20 further comprising instructions for causing the computer to modify the descriptor table to remove a portion of the segment that overlaps with the memory storing the translated code.

22. The computer program product of claim 20 further comprising instructions for causing the computer to modify the descriptor table to replace the segment with a substitute segment, which, when accessed, causes an exception to be generated.